

Comunicazione tra LabView e i Database

Redatto da: Gabriele Melani

Indice

Indice	2
Introduzione	3
A chi é indirizzato	3
Convenzioni e organizzazione	3
1. Acquisizione dati e database	4
2. Il linguaggio SQL utilizzato e i programmi necessari.....	4
3. Installazione e configurazione di MySQL	5
4. Implementare in LabView la comunicazione con il database creato	9
5. VI di esempio: eseguire una query singola su un database	14

Introduzione

In molti sistemi di acquisizione i dati rilevati devono essere memorizzati per elaborazioni successive; i database sono strumenti comodi per la memorizzazione dei dati.

In questo articolo vengono mostrati i passaggi fondamentali per realizzare la comunicazione tra LabView e un database, estrarre i dati significativi, effettuare elaborazioni e presentare i risultati tramite l'interfaccia di un VI.

A chi é indirizzato

A tutti coloro che desiderano interfacciare un database con LabView unendo la robustezza di una struttura a base di dati con le possibilità di elaborazione e presentazione che un VI può offrire.

Convenzioni e organizzazione

Per facilitare la lettura e l'apprendimento, sono inserite note distinte per il contorno esterno nero e al termine dei capitoli una sezione di riepilogo degli argomenti trattati.



Note



Riassunti di fine capitolo



Installazioni, connessioni e osservazioni sull' Hardware.



Collegamenti a risorse sul web

1. Acquisizione dati e database

Nella maggior parte dei sistemi di acquisizione dati dedicati, per esempio, al monitoraggio ambientale o alla misura di grandezze fisiche (elettriche, meccaniche, termiche...) che evolvono nel tempo, si presenta la necessità di memorizzare i valori rilevati per poter effettuare successive elaborazioni: i database sono strumenti comodi per l'archiviazione di dati.

Per "database" non si intende una semplice raccolta di dati: in alcuni manuali sul linguaggio SQL (la lingua dei database) si legge che "un database è una raccolta di record integrati, dove per record si intende un insieme di dati".

I database sono vere e proprie strutture organizzate e caratterizzate da relazioni tra i dati.

Esistono diversi tipi di database (indicati con la sigla DBMS, Database Management Systems) differenziati per il modo in cui vengono organizzati i record.

Tra i DBMS assumono un ruolo importante i database relazionali (indicati con la sigla RDBMS, Relational Database Management Systems) all'interno dei quali le informazioni sono organizzate in tabelle formate da righe e colonne. Ogni riga costituisce un record diverso e ogni colonna rappresenta un campo del record. Si consideri come esempio l'elenco del telefono: per ogni utente si possono leggere Nome, Cognome, Indirizzo, Numero di telefono.

Ogni utente è un record e ogni informazione descrittiva appartiene ad un campo.

Si immagini ora di eseguire una ricerca all'interno dell'elenco (per esempio tutti coloro che abitano in un paese o hanno un certo cognome) con l'obiettivo di estrarre solo i dati interessanti per i propri scopi.

In questo tipo di strumento righe e colonne sono relazioni tra dati: non conta la disposizione fisica dei dati ma la relazione che li unisce. I database RDBMS offrono almeno due differenze importanti rispetto ai DBMS:

- a. Consentono di aggiungere nuovi dati e nuove tabelle, eventualmente correlandole con una o più tabelle già esistenti, senza intervenire sui programmi applicativi che agiscono sulle tabelle già presenti.
- b. Sono database accessibili con SQL (Structured Query Language), strumento disponibile per le piattaforme hardware più usate con alcune differenze proprie del "dialetto" utilizzato.

2. Il linguaggio SQL utilizzato e i programmi necessari

Il linguaggio SQL può essere considerato un vero e proprio linguaggio di programmazione: tramite parole chiave e regole da seguire è possibile costruire frasi (chiamate "query") da utilizzare per chiedere al database dati particolari o modificare i dati già inseriti.

È un linguaggio "non procedurale": tramite SQL non si creano procedure per istruire un computer su ciò che deve fare ma si descrive al sistema ciò che si vuole ottenere.

Tramite le query il programmatore pone le domande al database e questo risponde con i dati richiesti.

Nel corso degli anni il linguaggio SQL ha subito cambiamenti ed evoluzioni; sono stati fatti tentativi per unificare i comandi in un unico linguaggio riconosciuto universalmente e se è vero che in parte ci si è riusciti (esistono standard riconosciuti per SQL) è anche vero che nel panorama informatico sono comparsi applicativi SQL caratterizzati da un proprio dialetto.

L'idea di fondo è sempre la stessa (eseguire operazioni su un database) ma la sintassi e le parole chiave del linguaggio sono diverse, a volte in maniera minima, tra un dialetto e l'altro.

In questo articolo verranno utilizzati due dialetti di SQL:

- Microsoft SQL
- MySQL

Si potrà notare come LabView sia in grado di comunicare con entrambe le versioni del linguaggio senza modifiche sostanziali al Block Diagram. Come avviene quando si mettono a confronto più linguaggi di programmazione, sarà sufficiente conoscere le regole proprie del singolo linguaggio per raggiungere gli obiettivi desiderati.

Prima di passare alla creazione del database e all'analisi dello steso tramite VI è indispensabile scoprire quali programmi ruotano attorno al linguaggio SQL.

Si è detto che le query costruite per comunicare con i database indicano al computer cosa si vuole ottenere e non come bisogna procedere per arrivare ai risultati voluti.

Nel computer ci deve essere un programma capace di trasformare le query scritte in linguaggio "ad alto livello" (più intuitivo per il programmatore) in istruzioni "linguaggio macchina" (comprensibili dal computer) per essere eseguite.

Sul proprio computer deve essere installato un "server" SQL capace di interpretare le query scritte con un particolare dialetto per trasformarle in istruzioni eseguibili dalla macchina.

Ogni server è in grado di convertire le istruzioni scritte seguendo una particolare sintassi.

Negli esempi riportati nell'articolo verranno utilizzati

- "Microsoft SQL Server" per lavorare con i database scritti in linguaggio Microsoft SQL
- "MySQL Server" per lavorare con i database scritti in linguaggio MySQL

Ogni server solitamente mette a disposizione del programmatore altri software da utilizzare per creare il database, inserire tabelle e dati, modificare i valori memorizzati e richiedere alcune informazioni particolari. Nello specifico

- "Microsoft SQL Server" mette a disposizione "Enterprise Manager" per visualizzare i database e "Query Analyzer" per costruire ed eseguire le query sui dati
- "MySQL Server" mette a disposizione "MySQL Administrator" per visualizzare i database e "MySQL Query Browser" per costruire ed eseguire le query sui dati

Nell'articolo verrà mostrata la configurazione dettagliata del server MySQL. Operazioni simili sono richieste durante l'installazione di altri server.

Sono stati scelti questi due dialetti SQL per mostrare come LabView si comporta quando si ha a che fare con versioni diverse del linguaggio SQL.

3. Installazione e configurazione di MySQL

Il pacchetto software MySQL può essere scaricato gratuitamente dal sito <http://www.mysql.com>; in particolare è possibile scaricare

- a. MySQL 5: installazione del server MySQL
- b. MySQL Administrator: configurazione parametri server e database
- c. MySQL Query Browser: manipolazione database attraverso query
- d. MySQL ODBC: driver per configurazione DSN del database

Il server permette di lavorare sul database MySQL sfruttando le query che possono essere create tramite "Query Browser" o altri programmi, nel nostro caso LabView.

Il driver ODBC permette la creazione di un collegamento al database all'interno delle impostazioni di Windows.



Sul sito <http://www.mysql.it> si possono trovare tutte le informazioni riguardanti il linguaggio MySQL.

Installare il server MySQL e i relativi programmi Administrator e Query Browser.
Durante l'installazione viene richiesta una password che poi sarà utilizzata per effettuare le operazioni sui database.

Al termine dell'installazione tra i programmi di Windows si trova una cartella nuova:



Fig. 1. Programmi MySQL installati

Avviare "MySQL System Tray Monitor": nella barra di Windows, vicino all'orologio compare un'icona di collegamento con i programmi MySQL installati.

Cliccare con il tasto destro del mouse sull'icona, selezionare "MySQL Administrator"; nella schermata di dialogo che si apre inserire i seguenti dati:



Fig. 2. Schermata di accesso al server

La password coincide con quella scelta durante l'installazione del Server MySQL.

Si apre la schermata di configurazione di MySQL: nella parte bassa della colonna di sinistra si può leggere "root" come utente collegato al server. Modificare questo valore inserendo un nuovo username: questo nome dovrà essere inserito al prossimo riavvio di "MySQL Administrator" nella casella "username".

Il server è stato configurato ed è pronto per essere avviato.

Per realizzare la comunicazione con il database è necessario configurare il DSN del database stesso.

In pratica il driver ODBC che è stato scaricato dal sito permette di creare un collegamento tra i programmi operativi e il database sul quale si effettueranno le operazioni di modifica e analisi.

Con DSN si intende "Data Source Name", nome di origine dati. Si tratta di un collegamento ad un database residente all'interno di una rete e accessibile da determinati programmi.

Controllare che il server MySQL sia in esecuzione (dal menù di "MySQL System Tray Monitor" raggiungibile dall'icona situata nella barra di Windows accertarsi che la prima voce sia "MySQL – Running". Se così non fosse nello stesso menù selezionare la voce "Start Instance").

In caso di problemi controllare le impostazioni del server avviando "MySQL Administrator".

Con il server in esecuzione si può procedere alla configurazione del DSN del database.

Forse è superfluo ma utile notare che prima di creare il riferimento al database è necessario creare il database stesso, costruendolo nuovo o importandolo da precedenti acquisizioni dati. Per creare un database nuovo in linguaggio MySQL è possibile utilizzare il "Query Browser" scrivendo le query di creazione tabelle e inserimento dati o mediante l'inserimento diretto di valori con la funzione "edit". Oppure si possono sfruttare altri software che mettono a disposizione dell'utente interfacce grafiche che semplificano il lavoro: uno strumento valido è EasyPHP (<http://www.easypHP.org>) che offre un'interfaccia grafica per la creazione dei database più semplice rispetto alla semplice interfaccia di inserimento query offerta da "MySQL Query Browser".

Aprire il "Pannello di Controllo" di Windows, selezionare "Strumenti di Amministrazione" e "Origine dati (ODBC)".

Si apre la finestra "Amministrazione origine dati ODBC"; selezionare l'elenco "DSN utente" e fare clic su "Aggiungi".

Nella schermata di selezione che si apre si possono leggere i nomi dei driver memorizzati nel sistema; scorrere l'elenco fino alla voce "MySQL ODBC 3.51 Driver" (la versione del driver può essere diversa a seconda del file che viene installato). Premere il pulsante "Fine".

Nella nuova finestra intitolata "Connector ODBC – Add data source name" inserire i parametri identificativi del database da analizzare:

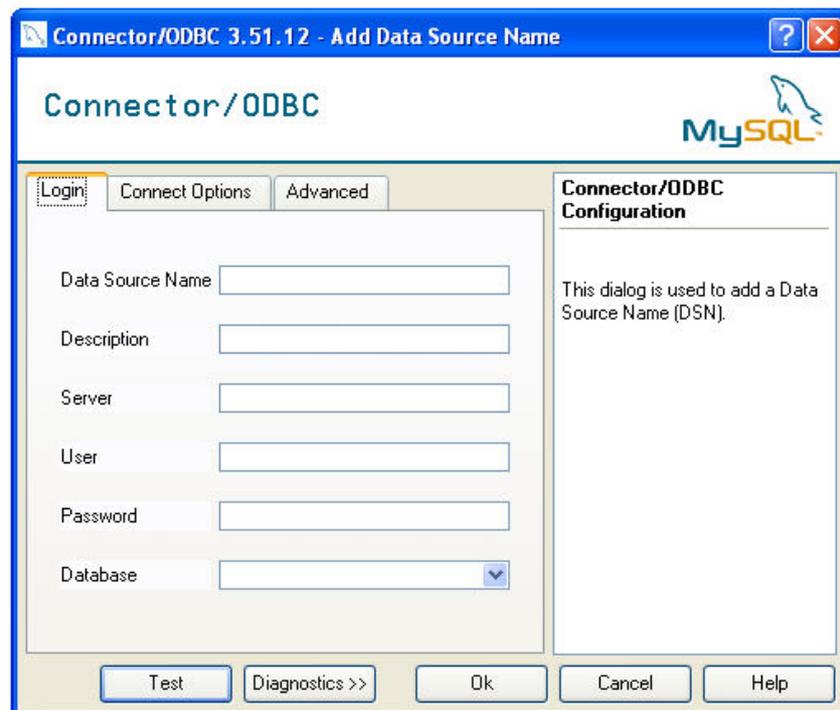


Fig. 3. Creazione di un nuovo DSN

- 1) **Data Source Name:** è il DSN del database, il nome che verrà utilizzato dalle applicazioni esterne (LabView) per accedere al database stesso. Inserire un nome a piacere (per questioni di stile di programmazione è bene inserire un nome attinente con il database associato).
- 2) **Description:** descrizione (facoltativa) del database in esame.

- 3) **Server:** nome del computer sul quale risiede fisicamente il database. Può essere lo stesso pc sul quale si sta lavorando (localhost) o un altro pc collegato tramite rete (indirizzo IP richiesto). Sul pc che ospita fisicamente il database deve essere in funzione il server MySQL.
- 4) **User:** UserID impostato in "MySQL Administrator" al posto di "root".
- 5) **Password:** parola codice scelta durante l'installazione del server MySQL.
- 6) **Database:** se i dati precedenti sono stati inseriti correttamente in questo menù è possibile visualizzare l'elenco dei database ai quali si può avere accesso.

Di seguito si riportano due situazioni esempio; nella schermata di selezione database inserire i seguenti dati se il database si trova sullo stesso pc sul quale è in esecuzione il server MySQL:

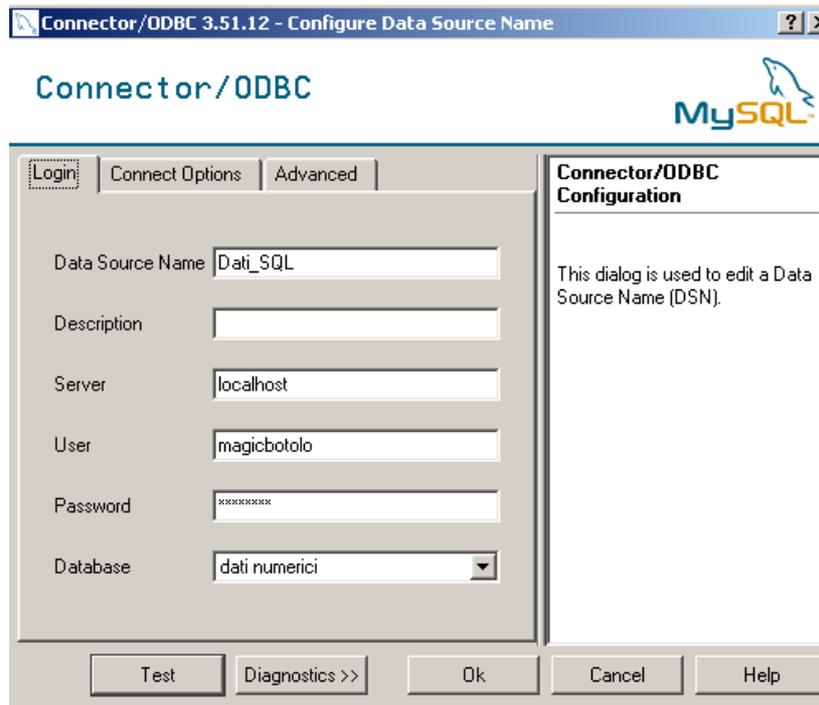


Fig. 4. Configurazione DSN per database presente sul pc

Se invece il database si trova su un altro pc collegato in rete inserire nella casella "Server" l'indirizzo IP di tale computer:

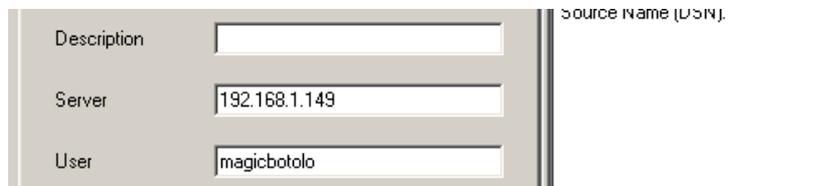


Fig. 5. Configurazione DSN per database presente su un pc in rete

Al termine della configurazione del DSN il database può essere richiamato da un programma operante sul computer con il nome identificativo scelto durante la configurazione.

Dopo aver inserito i dati, nella finestra di amministrazione dati del "Pannello di Controllo" compare il DSN appena configurato:

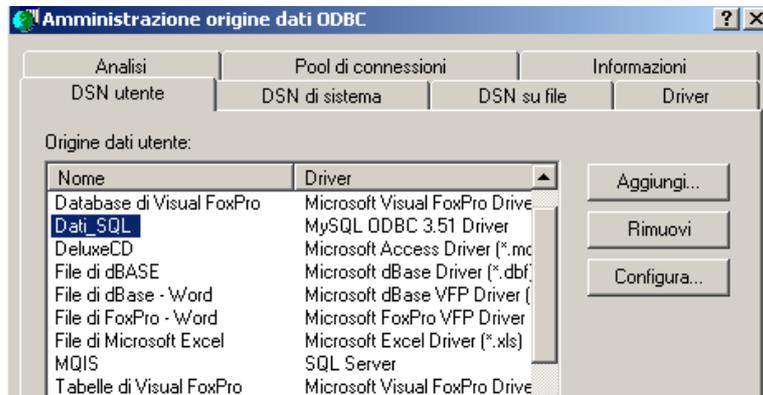


Fig. 6. Strumenti di amministrazione del Pannello di Controllo



La configurazione del DSN permette di accedere al database con programmi applicativi in grado di eseguire query; il nome scelto come DSN verrà utilizzato in LabView per richiamare il database.



Istruzioni per la configurazione del DSN sono disponibili all'indirizzo <http://freeasp.html.it/guide/lezione.asp?id=147>; qui si possono trovare altre informazioni utili per lavorare con i database MySQL.

4. Implementare in LabView la comunicazione con il database creato

Il linguaggio SQL può essere diviso in due grandi sezioni:

- Data Definition Language:** comandi usati per formare un database e creare tabelle; sono stati utilizzati durante la creazione del database tramite query o utilizzando un'interfaccia semplificata.
- Data Manipulation Language:** comandi usati per interrogare un database ed estrarre le informazioni volute.

In LabView sarà possibile lavorare con tutti i comandi che il linguaggio SQL mette a disposizione ricordandosi che ogni dialetto è caratterizzato da una propria sintassi: le parole chiave sono simili ma la struttura delle query cambia a seconda del dialetto SQL considerato.

In LabView è possibile creare frasi di interrogazione sfruttando i controlli di tipo "String". Una serie di stringhe collegate in successione può originare una query.

Una query semplice che può essere eseguita in MySQL per estrarre dati da una tabella di un database è la seguente:

```
SELECT * FROM <nome tabella>
```


La comunicazione con un database avviene per fasi successive; si inizia avviando il collegamento con la libreria ADODB. _Connection (per la connessione con gli oggetti ADO):

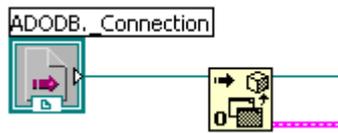


Fig. 10. Collegamento con libreria ADODB. _Connection

Una volta creato il collegamento con la libreria ADO si può lavorare con i "Property Node" e i "Method" di LabVIEW. In particolare verranno sfruttati i "Method" per interagire con il database ed eseguire le query.

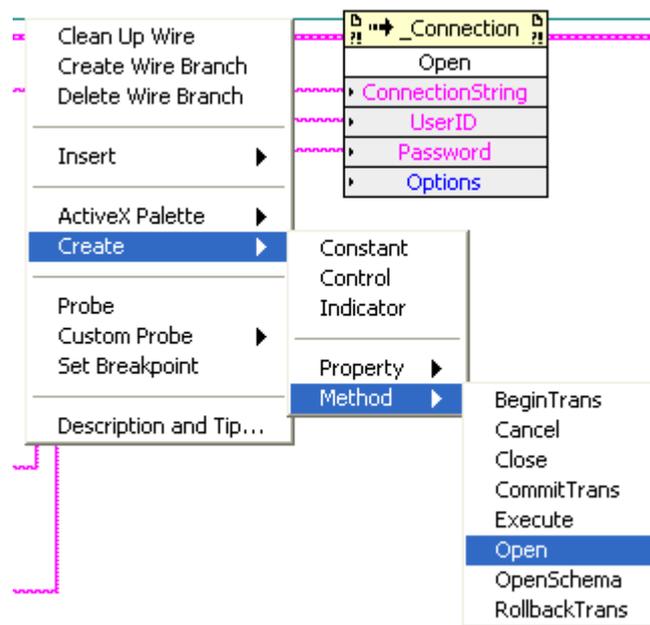


Fig. 11. Usare i "Method" di LabVIEW per lavorare sugli oggetti ADO

Il metodo **"open"** apre la comunicazione con il database da analizzare. Per fare questa operazione è necessario fornire in ingresso al metodo i dati del database, in particolare

- DSN del database
- UserID per l'accesso al database
- Password per l'accesso al database

Si ricorda che UserID e Password sono state scelte durante l'installazione del server o dopo l'accesso al programma di amministrazione (nel caso di MySQL la UserID è stata scelta nel programma "MySQL Administrator" e la Password durante l'installazione del server).

Nel VI il metodo "open" appare come mostrato di seguito:

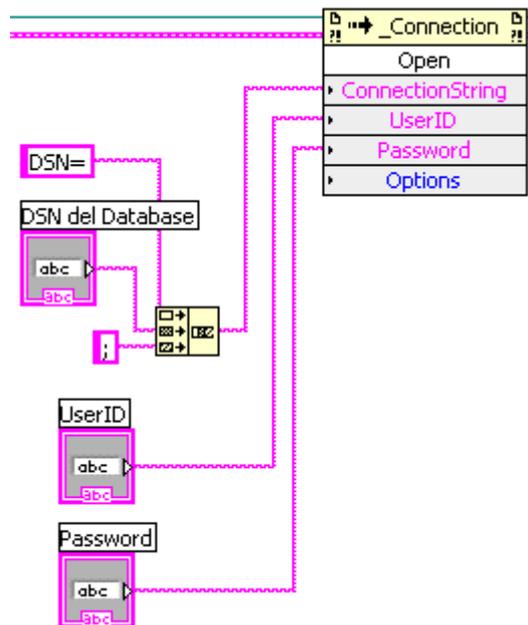


Fig. 12. Il metodo "open" con i dati del database

Come si può notare nella Fig. 12 al blocco del metodo "open" devono essere forniti UserID e Password sottoforma di stringhe e il DSN del database sottoforma di stringa costruita come di seguito:

DSN= <DSN scelto> ;

A questo punto ci si trova nella stessa situazione di quando si apre un file con un doppio clic del mouse o tramite la funzione "apri" del programma applicativo.

Dopo aver aperto il database si possono eseguire le query per modificarne la struttura o richiedere dati. Il metodo "execute" permette di lavorare sul database creando una query:

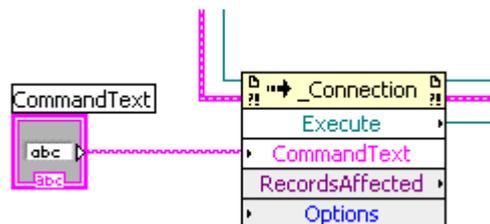
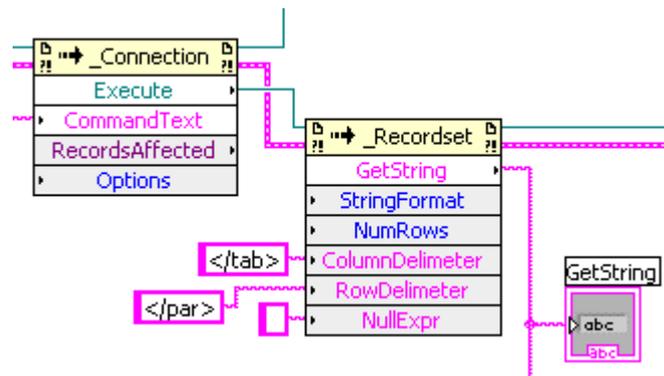


Fig. 13. Metodo "execute" per eseguire una query

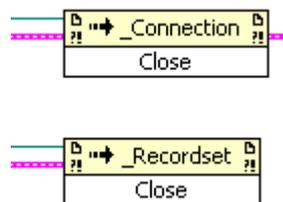
La stringa fornita come ingresso del metodo "execute" (nella Fig. 13 indicata con "Command Text") contiene la query con la quale si vuole analizzare il database; si può pensare ad una casella di testo abbastanza grande per contenere il testo della query scritta a mano dall'utente (Fig. 9) o possono essere preparati pulsanti collegati a query generali (visualizza tutto il contenuto di una tabella, esegui la media dei dati delle colonne della tabella e visualizza i risultati, visualizza i valori massimo e minimo di una certa colonna...). Il database risponde alla query con i dati richiesti: il metodo "getString" applicato al recordset (gruppo di dati) restituito dal metodo "execute" fornisce il risultato della query sottoforma di stringa.


Fig. 14. Metodo “getString” applicato al risultato della query

Questa stringa può essere formattata (Fig. 14, blocco “getString”) e successivamente manipolata per distribuire i record in una matrice e presentarli all’interno di una tabella.

Con gli indici della matrice è possibile lavorare sui singoli dati per effettuare le elaborazioni desiderate.

Al termine delle operazioni i riferimenti al database e al set di dati ottenuti tramite la query devono essere chiusi:


Fig. 15. Chiusura dei riferimenti al database e al recordset

Dal sito <http://www.jeffreytravis.com/lost/labsql.html> è possibile scaricare una libreria gratuita di comunicazione tra LabView e i database. Gli esempi all’interno di questa libreria lavorano con i blocchi presentati in questo articolo; inoltre è stata implementata la gestione degli errori che in questo articolo non è stata affrontata e sono stati realizzati blocchi per l’esecuzione di procedure particolari.

Quando si lavora con i database bisogna essere sicuri che le operazioni di apertura e inizializzazione della comunicazione diano esito positivo altrimenti le query porterebbero ad errori e problemi nel VI.

Si è incontrato un esempio di questo test di connettività durante la creazione del DSN nel “Pannello di Controllo” di Windows: l’ultima finestra (sia per MySQL che per Microsoft SQL Server) permette di eseguire una prova di accesso al database prima di confermare i dati.



La comunicazione tra LabView e i database avviene per fasi successive: tramite la funzione ADODB. _Connection viene richiamato il database. Con il metodo “open” si accede al database, con il metodo “execute” viene elaborata la query e i dati ottenuti come risultato possono essere formattati (con il metodo “getString”) e visualizzati in forma tabellare.



Al sito <http://www.jeffreytravis.com/lost/labsql.html> è possibile scaricare una libreria gratuita di comunicazione tra LabView e i database.

5. VI di esempio: eseguire una query singola su un database

Come esempio viene presentato un VI che esegue una singola query su un database e presenta i dati ottenuti in forma matriciale:

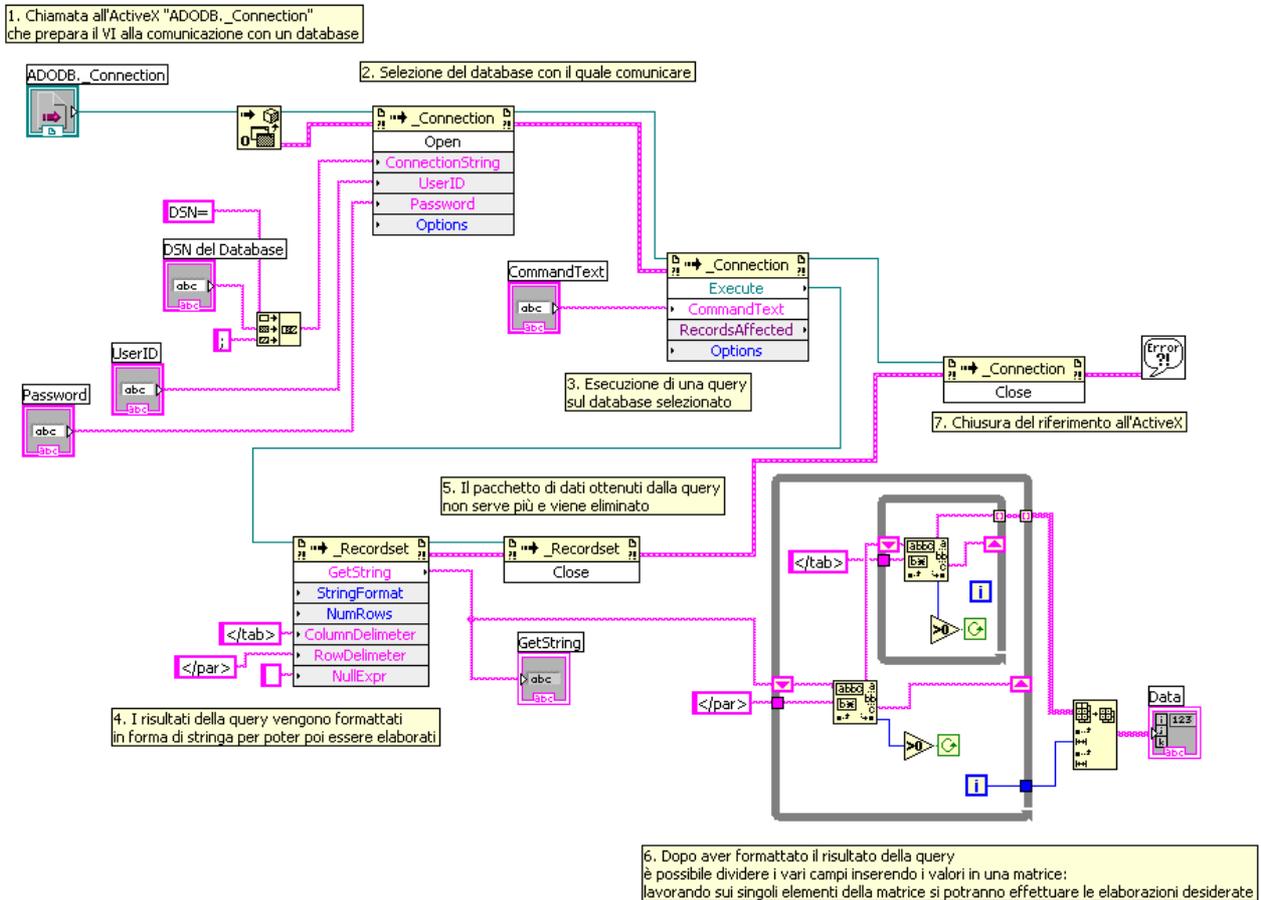


Fig. 16. Block Diagram

In Fig. 16 si possono vedere i metodi applicati al database collegati in sequenza:

1. Apertura ActiveX
2. Scelta del database e connessione
3. Inserimento query ed elaborazione
4. Formattazione dei risultati ottenuti e chiusura del recordset
5. Chiusura del riferimento al database analizzato

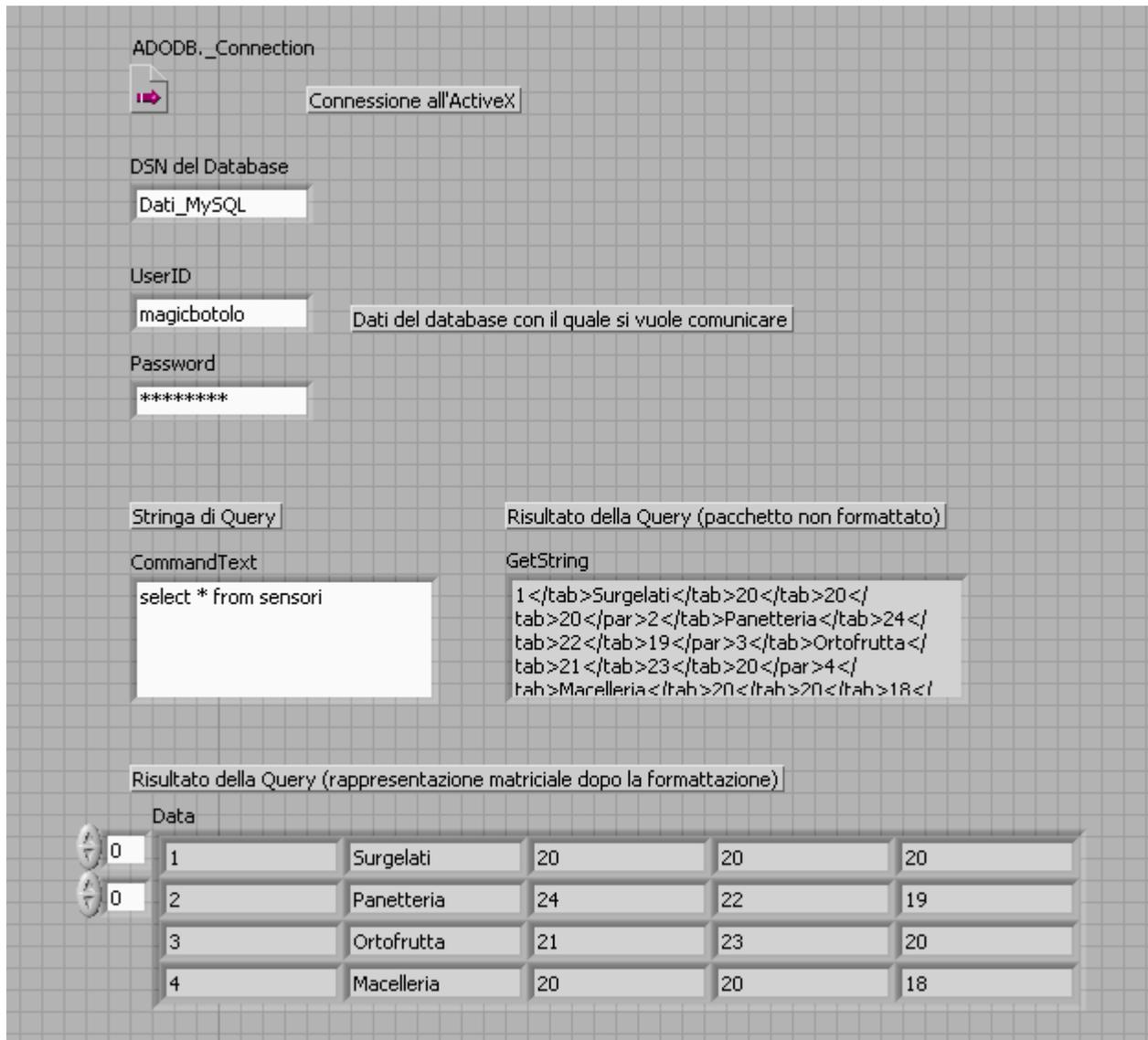


Fig. 17. Front Panel

In Fig. 17 si possono notare i valori forniti ai blocchi del block diagram (Fig. 16):

- DSN, UserID, Password per l'accesso al database
- Query di interrogazione
- Risultato dell'interrogazione seguita

È interessante osservare il contenuto della casella di testo denominata "getstring" ottenuta come risultato del metodo "execute". I risultati della query posta al database vengono restituiti sottoforma di stringa alfanumerica. Questa stringa può essere formattata in base alle proprie esigenze per essere successivamente elaborata e suddivisa nei diversi campi al fine di creare una tabella (i dati all'interno della tabella "data" coincidono con i dati presenti nella casella di testo "getstring" divisi per campo di appartenenza). Utilizzando le celle della tabella (lavorando sugli indici di riga e colonna) possono essere isolati singoli valori per eseguire operazioni particolari quali media, massimo, minimo... I dati ottenuti possono poi essere visualizzati in forma grafica passando in ingresso al "chart" / "graph" un vettore creato con i dati estratti dalla tabella.

Di seguito viene riportata una soluzione ottimizzata per la formattazione dei dati ottenuti tramite query (proposta da Alessandro Ricco, *ILVG Admin*):

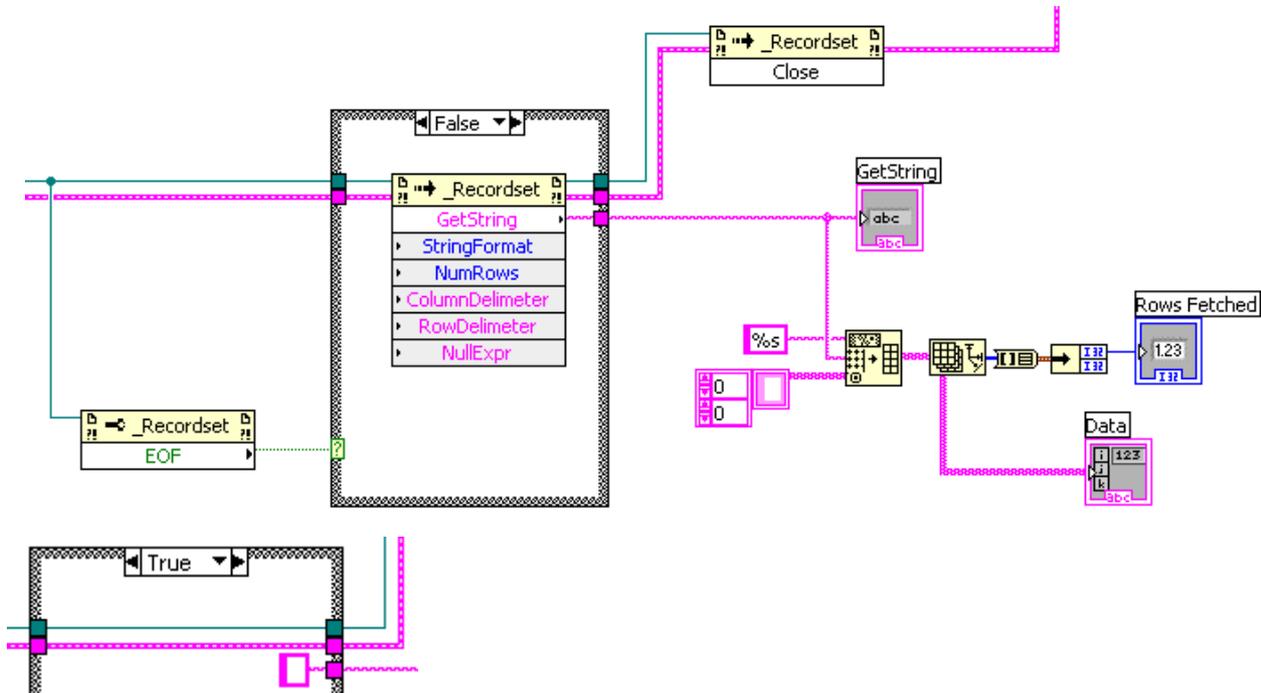


Fig. 18. Formattazione alternativa dei risultati

Seguendo questa procedura i risultati della query vengono formattati secondo la seguente struttura:

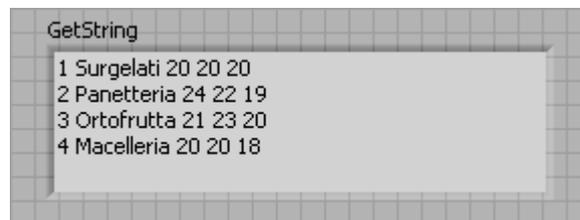


Fig. 19. Casella “getString” contenente il risultato della query

E se invece il database è stato realizzato con Microsoft SQL?

Il VI riportato in Fig. 16 non subisce variazioni. Quello che cambia è la query: il dialetto compreso da “Microsoft SQL Server” segue regole diverse da MySQL. Modificando la query in relazione al dialetto usato si ottengono i risultati desiderati.

Questo significa che se la query di interrogazione è stata inserita “a mano libera” dall’utente è sufficiente cancellarla e riscriverla con la sintassi corretta mentre in caso di query costruite con blocchi “String” deve essere modificato l’ordine di collegamento dei blocchi stessi.

Di seguito viene mostrata una query complessa realizzata in MySQL e in Microsoft SQL: l’ordine delle parole chiave nella frase è diverso e, di conseguenza, è diverso l’ordine di collegamento dei blocchi costituenti la query in LabView. Il resto del VI rimane uguale.

MySQL

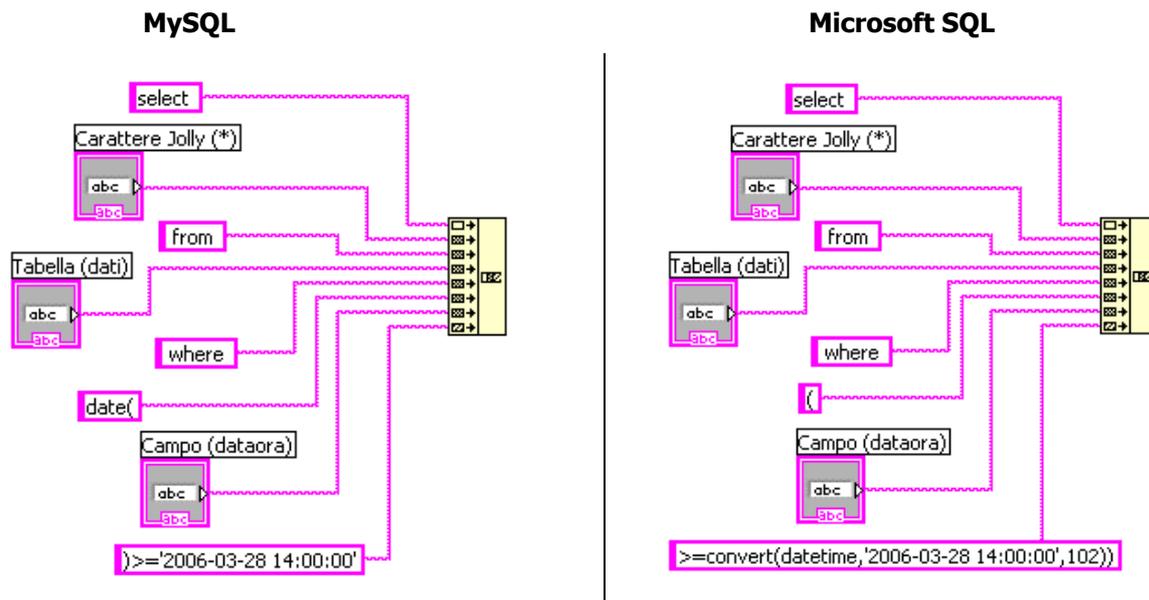
```
SELECT * FROM dati WHERE date(dataora)>='2006-03-28 14:00:00'
```

Microsoft SQL

```
SELECT * FROM dati WHERE (dataora >= CONVERT(DATETIME, '2006-03-28 14:00:00', 102))
```

In entrambi i casi la query estrae dalla tabella "dati" del database i record registrati dopo le ore 14.00 del 28 Marzo 2006 (data e ora di acquisizione del dato sono memorizzati nel campo "dataora" della tabella "dati").

In LabView le query assumono l'aspetto seguente:



Quindi si può concludere che LabView può essere utilizzato per lavorare con diversi tipi di database senza modificare interamente il VI ma adattandolo al dialetto SQL considerato.



All'indirizzo

<http://zone.ni.com/devzone/conceptd.nsf/webmain/FA0A99E9BD96C8E98625713D007167EC#2> si può leggere il manuale di "LabView Database Connectivity Toolkit", un tool integrativo di LabView contenente le librerie di comunicazione con i database. Può essere interessante sfogliare il manuale per conoscere altre nozioni riguardanti l'interazione tra LabView e i database.

Altre informazioni sui database e l'interazione con LabView si possono leggere all'indirizzo

http://www.lavag.org/archives/ON/files/Using_MySQL_with_LabVIEW.pdf.

Dal sito ILVG, nella sezione articoli, è possibile scaricare il documento preparato da Nicola Bavarone dal titolo "Database e LabView" (<http://www.ilvg.it/forum/download.php?id=299>). In questo articolo viene trattata la comunicazione tra LabView e i database Access (tecnologie DAO e ADO) e si può leggere una breve presentazione del toolkit della National Instrument per l'accesso ai Database.